(2004). In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation*. ISBN: 1-930638-35-3.

A Cognitively Plausible Model of Language Comprehension

Jerry T. Ball
Human Effectiveness Directorate, Air Force Research Laboratory
6530 S. Kent St., Mesa, AZ 85212-6061
Jerry.Ball@mesa.afmc.af.mil
www.DoubleRTheory.com

Keywords:
Language Comprehension, Natural Language Processing (NLP), Cognitive Modeling

ABSTRACT: *Double R Model (Referential and Relational Model) is a model of language comprehension intended for use in the development of software agents with NLP capabilities. Double R Model is fairly unique in adopting a cognitively plausible approach to modeling language comprehension, while at the same time attempting to support the development of large-scale, functional models. The key claim is that adhering to well-established cognitive constraints on language comprehension serves to prune the search space for possible solutions and may actually facilitate the development of functional NLP systems.*

1. Introduction

Double R Model (Referential and Relational Model) is a model of language comprehension intended for use in the development of software agents with NLP capabilities. Double R Model is fairly unique in adopting a cognitively plausible approach to modeling language comprehension, while at the same time attempting to support the development of large-scale models. That is, although Double R Model adheres to well-established cognitive constraints on language comprehension, it is not focused on empirical validation of specific psycholinguistic phenomena. On the other hand, Double R Model does not make use of AI and Computational Linguistic techniques like full unification and unlimited backtracking which are cognitively implausible, but often adopted uncritically in NLP systems without a focus on cognitive plausibility. The key claim is that adhering to well-established cognitive constraints on language comprehension serves to prune the search space for possible solutions and may actually facilitate the development of functional NLP systems. Given the inherently human nature of language and the huge capacity and extreme complexity of linguistic knowledge, language comprehension systems which fail to consider well-established cognitive constraints may prove inadequate just where they diverge most from human linguistic behavior.

2. Double R Grammar

Double R Grammar [1] is the Cognitive Linguistic theory [2,3,4] underlying Double R Model. In Cognitive Linguistics, all grammatical elements have a semantic basis, including parts of speech, grammatical markers, phrases and clauses. Our understanding of language is embodied and based on experience in the world [5]. Categorization is a key element of linguistic knowledge. Categories are seldom absolute, exhibiting, instead, effects of prototypicality, base level categories [6], family resemblance [7], fuzzy boundaries, radial structure and the like [8]. Our linguistic capabilities derive from basic cognitive capabilities—there is no autonomous syntactic component separate from the rest of cognition. Knowledge of language is for the most part learned and not innate. Abstract linguistic categories (e.g. noun, verb, nominal, clause) are learned on the basis of experience with multiple instances of words and expressions which are members of these categories, with the categories being abstracted and generalized from experience. Also learned are schemas which abstract away from the relationships between linguistic categories. Over the course of a lifetime, humans acquire a large stock of schemas at multiple levels of abstraction and generalization representing knowledge of language and supporting language comprehension.

Two key dimensions of meaning that get grammatically encoded are referential and relational meaning. Consider the expressions

1. The aircraft on the runway
2. The aircraft is on the runway

These two expressions have essentially the same relational meaning. They both express the relation “on” existing between “an aircraft” and “a runway”. However, their referential meaning is significantly different. The first expression, as a whole, refers to an object and is called an **object referring expression**. In referring to an object, the first expression uses the determiner "the" to **specify** that the object is salient in the context of use of the expression (and may have previously been referred to). The first expression also uses the word "aircraft" to indicate the type of object being referred to, with "aircraft" functioning as the **head** of the expression. Further, the phrase "on the runway" refers to a location with respect to which the object can be identified and functions as a **modifier** in the expression. In referring to a location, the expression "on the runway" refers to a second object "the runway" and indicates the location of the first object with respect to the second object. The relation "on" functions as the **relational head** with the object referring expression "the runway" functioning as a **complement**. In the first expression, the relational meaning of "on" is subordinated to referential meaning with the modifying function of "on the runway" dominating the relational meaning of "on". That is, although "on" is the relational head of the prepositional phrase "on the runway", it is not the head of the overall expression and does not determine the relational type of that expression. A graphical representation of this expression is shown below:

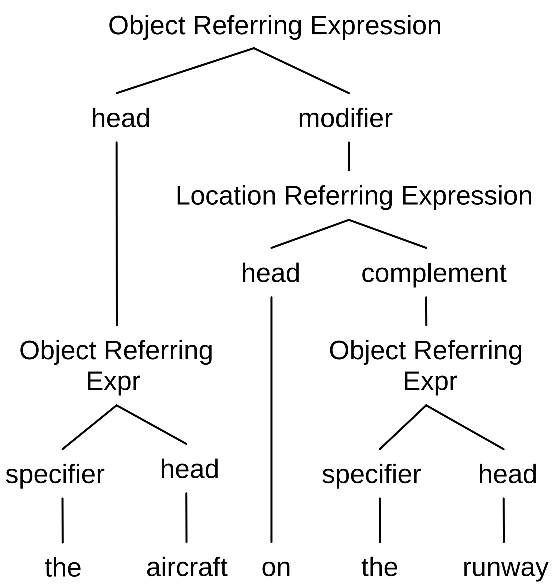

Figure 1: an object referring expression

The second expression refers to a situation and is called a **situation referring expression**. This expression uses the auxiliary "is" to provide a temporal specification for the situation, fulfilling a referential function similar to that of the determiner "the" in "the aircraft" and "the runway". The relational meaning of the second expression is about "being on" and not just "being", with "on" functioning as the relational head. The relational head of a situation referring expression is called a **predicate**—reflecting the assertional function of the relational head. Note that "on" in the first expression is not functioning as a predicate, since it is presupposed and not asserted. In the second expression, the object referring expression "the aircraft" functions as the subject (argument) of "being on" with "the runway" functioning as the object (argument). Referentially, there is also a reference to a location "on the runway", which competes with the expression of the relational meaning of "on" as reflected in:

3. What is the aircraft on?
4. Where is the aircraft?

where 3 highlights the relation "on" in asking about the object of that relation and 4 highlights the reference to a location using "where" to do so. A graphical representation of the relational meaning of this expression is shown below:

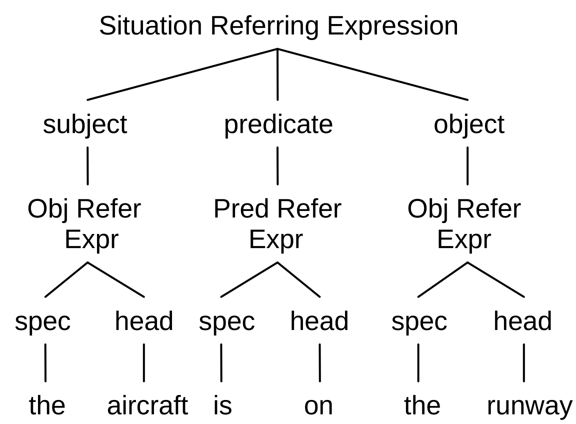

Figure 2: a situation referring expression

In this representation, the (predicate) specifier "is" combines with the relation "on" to form a predicate referring expression which functions as the head (i.e. predicate) of the situation referring expression.

The terms **specifier**, **head**, **modifier** and **complement** are borrowed from X-Bar Theory [9]. It is acknowledged that X-Bar Theory captures an important grammatical generalization, with the distinction between specifiers and modifiers representing a significant advance, but X-Bar theory is in need of semantic motivation [10], which, when provided, necessitates certain modifications to the theory. In particular, Double R Grammar presents a bi-polar theory of nominal and clause structure consisting of a referential pole and a relational pole—with the specifier functioning as the locus of the referential pole and the head functioning as the locus of the relational pole. X-Bar theory is a uni-polar theory with the head functioning as the only pole, which leads to confusion about what the head of an expression should be. For example, in a nominal like “the jet” it is now common in X-Bar Theory to treat “the” as the head of a DP (Determiner Phase) taking “jet” as an NP (Noun Phrase) complement. However, “jet”, by itself, isn’t even an NP, and treating “the” as the head instead of “jet” violates any traditional notion of what a head is, leading McCawley to lament that in X-Bar Theory “…all sorts of things…get represented as heads of things they aren’t heads of” [11].

In Double R Grammar’s bi-polar theory, the specifier determines the referential type of a referring expression (corresponding to a maximal projection in X-Bar theory), whereas the head determines the relational type of the expression (where relational type encompasses non-relational objects). Consider the referring expression

3. The attack

in which the (object) specifier “the” determines the expression to be an object referring expression, whereas, the head “attack” determines the expression to be a type of action. In this expression, the specifier has the effect of objectifying the action expressed by “attack” and allowing it to be referred to as though it were an object. Note that since the inherent meaning of “attack” is not affected, only its grammatical function, there is no need to assume that the part of speech of “attack” is a noun instead of a verb in this expression. And if we allow verbs (especially action verbs) to function as heads of object referring expressions (nominals), then one of the primary syntactic arguments against the meaning based definition for parts of speech is nullified [12]. More generally, verbs, adjectives, adverbs and prepositions are types of relations which evoke relational schemas containing arguments, whereas nouns are non-relational and do not evoke such schemas [1,3]. Relations may head object referring expressions, but when they do, they are objectified and their arguments are typically suppressed. By comparison, in X-Bar theory, the head of a NP is necessarily a noun since the head always determines (or projects) the type of the expression. However, as argued in [12], words of many different parts of speech and numerous phrase and clause types can head NPs.

In sum, Double R Grammar’s semantically based bi-polar theory of nominal and clause structure represents an improvement over syntactically based uni-polar theories like X-Bar Theory. Adding a specifier as the locus of referential meaning is an extension of Langacker’s [3] conception of nominals and clauses, with the specifier functioning as the locus of Langacker’s **grounding predication**.

3. Double R Process

Double R Process is the theory of language processing underlying Double R Model. Double R Process is intended to be a cognitively plausible processing mechanism for constructing integrated representations of referential and relational meaning. In this respect, it follows the research approach of the Competition Model [13] and accepts many of the assumptions of that model. For example, the Competition Model is functionalist in its linguistic orientation—with functionalism sharing many of the theoretical insights of Cognitive Linguistics and contrasting with mainstream generative theories (e.g. Transformational Grammar, Government and Binding Theory). In the Competition Model, “the underlying linguistic representation is strongly lexicalist, using a parser that is bottom-up and cue-driven to construct a dependency graph, rather than a standard parse tree”. During language processing “each lexical item sets up expectations for other lexical items…The processor attempts to develop a relational structure…Semantic interpretation works upon this structure interactively…The inclusion of arguments on the basis of semantic cues is a tentative process…The parser is basically driven by the attempt to instantiate the arguments of each predicate” [all quotes from 13].

Double R Process is highly interactive. Meaning representations are constructed directly from input texts. There is no separate syntactic analysis that feeds a semantic interpretation component. The processing mechanism is driven by the input text in a largely bottom-up, lexically driven manner. There is no top-down assumption that a privileged linguistic constituent like the sentence will occur (unlike [14]).

In Double R Process there is no phrase structure grammar and no top-down control mechanism. How then does Double R Process construct representations of input text? Operating on the text from left to right, schemas corresponding to lexical items are activated. For those lexical items which are relational or referential, these schemas establish expectations which both determine the possible structures and drive the processing mechanism. A short-term working memory [15,16] is available for storing (or keeping active) arguments which have yet to be integrated into a relational structure, partially instantiated relational and referential structures, and completed structures. If a relational entity is encountered which expects to find an argument to its left in the input text then that argument is assumed to be available in short-term working memory. If the relational entity expects to find an argument to its right in the input text, then the relational entity is stored (or kept active) in short-term working memory as a partially completed structure and waits for the occurrence of the appropriate argument. When that argument is encountered it is instantiated into the relational structure in short-term working memory. In the case where the text contains a profiled or otherwise salient non-relational entity (e.g. the subject), that entity may be made separately available in short-term working memory. Otherwise, non-profiled and non-salient arguments are incorporated into relational structures and are not separately available. This keeps the number of separate linguistic units which must be maintained in short-term working memory to a minimum. Key components of the processing mechanism are the activation, selection and integration of schemas associated with the referential or relational units in the input text. These schemas set up expectations which drive the processing mechanism and they also function as the key determiners of the structure of the input text.

It is assumed that knowledge of language consists largely in the availability of schemas at multiple levels of abstraction and generalization, with more concrete schemas carrying most of the burden for language processing. But how might these schemas be organized, and how might they be accessed in language processing? It is assumed that these schemas are organized in the form of an associative network over which a spreading activation process operates [17,18]. As a piece of text is processed, schemas containing representations of lexical items which correspond to lexical items in the input text will be activated and will in turn activate associated schemas to some degree. Given this spreading activation mechanism, it follows that those schemas which most closely correspond to the input text will be most strongly activated. For the most part, these schemas will be concrete and lexically laden. Very abstract schemas which contain no lexical items can only be indirectly activated since they have no direct correspondence to the input text. Further, a schema may be activated despite the fact that it does not correspond exactly to the input text. This fact makes it possible for the system to deal with degraded or erroneous input, although in general the closer the correspondence between the input text and a schema, the higher the activation of that schema. In addition to the activation mechanism, there must be some selection mechanism for choosing among the activated schemas. In the simplest case this mechanism may simply select the most highly activated schema, and this selection process may be automatic. But selection of a particular schema should not preclude subsequent change in the context of new information and it may also be the case that more than one schema may be selected under certain circumstances (e.g., in the case of puns and double entendres). Thus, the selection process is both tentative (subject to revision) and preference based [19].

Once selected, a schema must be integrated with the preceding linguistic context. The basic mechanism for integration is **elaboration** [3]. A relational schema typically provides only an abstract, limited representation of its arguments since those arguments vary from text to text. For example, the schema **|subject attack object|** provides only the limited information that there is a subject argument that is an object referring expression which typically occurs before **attack** in the input stream, and that there is an object argument, also an object referring expression, typically occurring after **attack** in the input stream. When the word “attacked” is processed, this schema is likely to be activated and selected. Note that **attack** is itself schematic for the different forms that “attack” takes. The integration of this schema with the preceding linguistic context depends on the availability of an object referring expression capable of elaborating the subject argument (assuming the subject argument is integrated at this point).

A consideration of the application of schemas to the processing of language offers several insights. If high-level abstract schemas like **|subject predicate object|** drive the processing mechanism, then the processor could be said to be grammar driven. On the other hand, if low-level schemas with specific lexical items drive the processing mechanism, then the processing mechanism is essentially lexically driven. As noted above, it is an assumption of Double R Process, in agreement with Langacker [2], that more specific schemas have “priority” over more abstract schemas in normal processing, and that most of the knowledge needed for the processing of familiar expressions has been lexicalized in the form of schemas containing specific lexical items. In unusual situations, abstract schemas may assume greater importance. For example, second language learners who are explicitly taught the grammar of a language may rely on more abstract schemas than native speakers of the language—not only as a result of instruction, but because they may lack the more specific schemas available to native speakers.

It is important to note that schemas set up expectations for the occurrence of various elements, but do not preclude the occurrence of other elements. That is, the application of a schema does not require the exact matching of the elements of the schema with elements of the input text. In this regard, schemas do not behave like the rules of a phrase structure grammar. For example, the **|subject refuel object|** schema sets up expectations for the occurrence of a subject before “refuel” and an object after it, but does not preclude the occurrence of an adverb or prepositional phrase modifier in any input text to which the schema corresponds. Thus, the schema is entirely consistent with the sentence

4. She always refuels fighters on Friday

The occurrence of the adverb “always” and the prepositional phrase “on Friday” in this sentence activate additional schemas which must be integrated with the |**subject refuel object**| schema during the construction of a representation for this sentence. There is no need for a schema to specify all possible elements which can occur at all possible positions in the schema so long as a mechanism for integrating multiple schemas exists. On the other hand, the existence of schemas which are completely specified with regard to their lexical content and are only weakly integrable with other schemas is not precluded (e.g. schemas corresponding to idiomatic and formulaic expressions). In general, abstract schemas will be easily integrated with other schemas, whereas, concrete schemas will be less easily integrated.

The **Activation-Selection-Integration** mechanism which is the cornerstone of Double R Process involves one automatic and two control processes: (a) an automatic spreading activation process, (b) a control process for selecting activated schemas from long-term memory and placing them (or making them active) in short-term working memory, and (c) a control process for integrating selected schemas in short-term working memory. Activation-Selection-Integration is a clausal level variation of the discourse based **Construction-Integration** (C-I) mechanism of C-I theory [16]. C-I theory differs in that both construction and integration are highly automated processes. Construction involves the unregulated activation of all concepts associated with an input. Integration is a settling out or constraint-satisfaction process involving excitatory and inhibitory links between activated concepts. Selection, to the extent that it is modeled in C-I theory, occurs as a result of the settling out process during integration. In Double R Process, selection is a control process which provides functionality similar to the automated inhibitory links in C-I Theory. Although described as a control process, selection has many automated subcomponents and in the limiting case (routine processing) may be fully automatic. A recent variant of C-I theory, the **Immersed Experiencer Framework** (IEF) [20], “distinguishes three component processes of language comprehension: activation, construal and integration. Activation operates at the word level, construal at the clause level, and integration at the discourse level.” The notion of construal is introduced to reflect the creation of a perceptually based, mental simulation of an event corresponding to a clausal input. Construal in IEF corresponds to the perceptual grounding of referring expressions in a **situation model** in Double R Theory, a topic which is discussed in [1].

The control process of integrating selected schemas in short-term working memory can be described algorithmically in terms of the individual processing steps required to integrate the schemas. This process will be illustrated by walking through the steps involved in the processing of the following English sentence:

5. The extremely old Navy captain likes to sleep a lot

The processing of this sentence begins with the activation and selection of a schema corresponding to the first lexical item. The word “the” is identified and a referential schema which reflects its typical function as a specifier which combines with a head to form an object referring expression is selected:

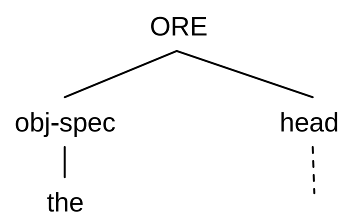

Figure 3: an object specifier schema

In this representation, ORE stands for object referring expression, obj-spec stands for object specifier, and the dashed line extending from head indicates that this component has not yet been elaborated. This schema is equivalent to a schema of the form |**obj-spec head**| except that the type of the composite expression (object referring expression) is explicitly represented and the object specifier function is elaborated by the determiner “the”. Part of speech information is left out of the schema since the focus is on identifying the functional role that the lexical item “the” takes on. Since the head is expected to occur after the specifier in this schema, the specifier must await the appearance of this head before combining with it to form an object referring expression. The specifier schema is retained in short-term working memory with its head unelaborated and the processing of this specifier is temporarily halted.

The processing of the next lexical item begins. The word “extremely” is identified and determined to be an adverb. In the context of an object specifier schema, “extremely” is presumed to be functioning as a relation modifier as represented by the following schema:

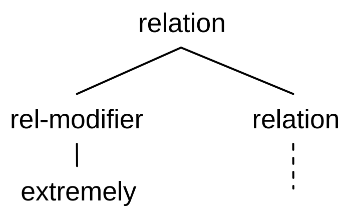

Figure 4: a relation modifier schema

According to this schema, the relation that "extremely" modifies typically occurs to its right in the input stream (at least in the context of an object specifier) and processing of the relation modifier is temporarily halted.

The processing of the next lexical item begins. The word "old" is identified and determined to be an adjective. In the context of an object specifier and relation modifier, "old" is presumed to be functioning as an object modifier and the following schema is placed in short-term working memory:

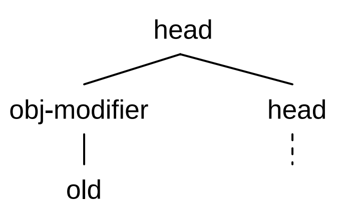

Figure 5: an object modifier schema

Note that there are three separate schemas in short-term working memory at this point. Since "old" is a relation, its schema can be integrated with the relation modifier schema giving:

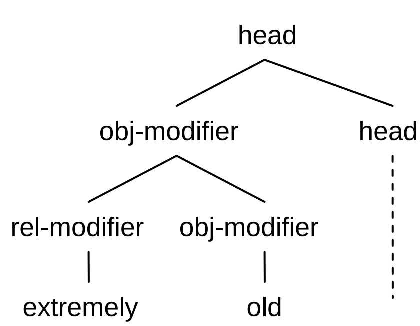

Figure 6: integrating a relation modifier with an object modifier

The head that occurs with "old" typically occurs to its right and the processing of "old" is temporarily halted. Note that whereas the schema for "extremely" is integrated with the schema for "old" prior to the integration of a head into the "old" schema, this schema is not integrated with the specifier schema for "the" at this point—since integration with the specifier is presumed to require a fully elaborated head.

Processing continues with the next lexical item. The word "Navy" is identified and determined to be a noun. In the context of an object specifier and object modifier schema, "Navy" is determined to be a head that elaborates the head function in the schema for "old" giving:

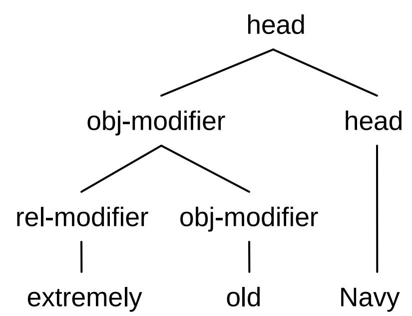

Figure 7: integrating a head with an object modifier

This modified head then elaborates the head function of the specifier schema of "the" giving an object referring expression:

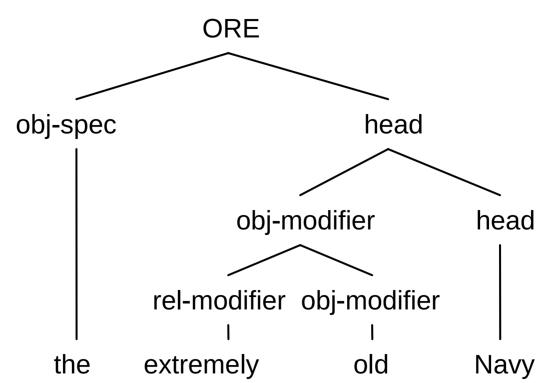

Figure 8: integrating a specifier with a head

At this point in processing, there is a single object referring expression which is a fully elaborated schema in short-term working memory. Object referring expressions are non-relational and do not typically establish expectations for other linguistic units.

Processing continues with the next lexical item. The word "captain" is identified and determined to be a noun. In the context of an object referring expression, "captain" is determined to be the head of that expression, displacing the current head and making it a modifier leading to:

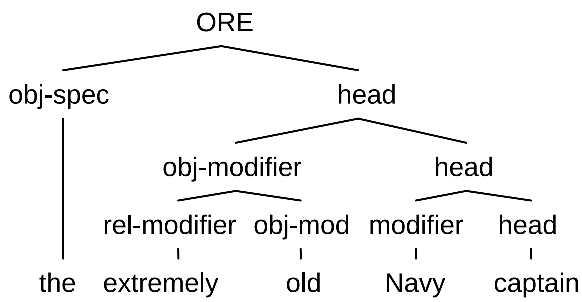

Figure 9: accommodating a second head noun

The basic mechanism for processing “captain” in this context is to modify an existing schema to **accommodate** it. There is a learned production (or series of productions) available to support such accommodation and it is not a question of backtracking and trying different alternatives until a correct structure is obtained. The major advantage of accommodation over backtracking is that the full current context is available to support accommodation, whereas in backtracking the context is typically unraveled and only the fact that the current choice is incorrect is typically available to the processing mechanism. That is, the context which forced the backtracking is not available to guide the selection of an appropriate structure.

Following the accommodation of “captain”, processing continues with the next lexical item. The word “likes” is identified and determined to be a present tense verb which determines a predicate referring expression functioning as a predicate in a situation referring expression with two participants, a subject and an object:

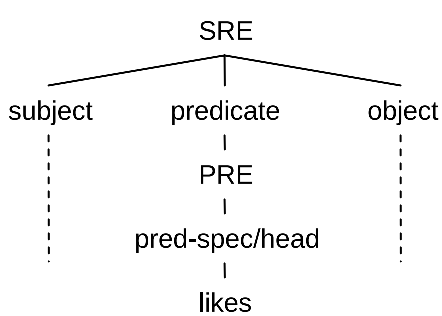

Figure 10: a relational head schema

where SRE is situation referring expression, PRE is predicate referring expression, and pred-spec is predicate specifier. Note that “likes” functions as both a predicate specifier and head. The present tense marking of “likes” provides morphological support for the predicate specifier function. In the context of an object referring expression, the object referring expression elaborates the subject of the situation referring expression giving:

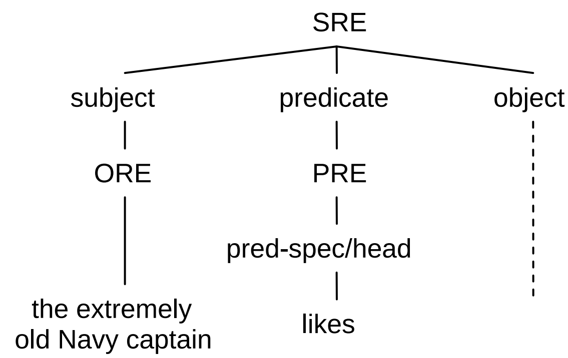

Figure 11: elaborating the subject argument

Note the assumption that the processing mechanism immediately elaborates the subject and does not wait until after the processing of the object to do so. The common assumption that a sentence is divided into a subject (NP) and predicate (VP including any objects and not just the V) is inconsistent with the elaboration of the subject at this point. However, the asymmetry of subjects and objects and the salience of the subject relative to the object is assumed not to be reflected in the processing mechanism in terms of the delayed elaboration of the subject relative to the object. Instead, the salience of the subject may be a result of Gernsbacher’s empirically supported principle of **First Mention** [21] and/or other pragmatic considerations (retaining the subject separately in short term working memory).

Once the subject is elaborated, the processing of the schema for “likes” is temporarily halted and processing continues with the next lexical item. The highly ambiguous word “to” is identified and its function in the current context is not immediately determined. The subsequent context is needed to resolve the ambiguity. The next lexical item is processed and “sleep” is identified and determined to be a verb. In the context of “to”, the non-finite, intransitive predicate referring expression “to sleep” is identified and its schema is retrieved and made available:

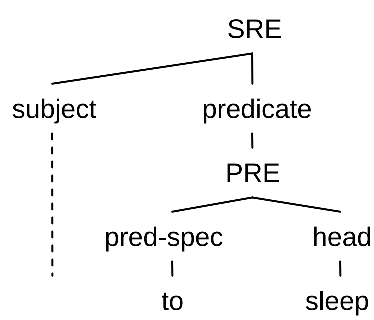

Figure 12: another relational head schema

This schema is integrated with the schema for "likes" elaborating the object argument, where this elaboration involves accommodating the clausal complement "to sleep" by converting the object argument to a clausal complement (which could be further specialized as an infinitive clause). That conversion may involve the retrieval of a schema for "likes" which includes a clausal complement and the integration of that schema with the existing schema for "likes", or it may more simply involve the conversion of the object argument of the existing schema to a clausal complement:

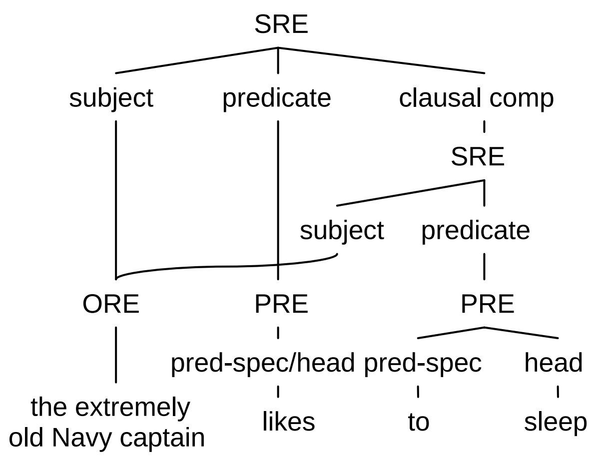

Figure 13: integrating two relational head schemas

Another part of the integration involves identifying the subject of "to sleep" with the subject of "likes". The result of this integration is a fully elaborated situation referring expression.

Following the integration of the "likes" and "to sleep" schemas, processing continues with the next lexical item. The word "a" is identified and a specifier schema is retrieved. Then the word "lot" is identified and in the context of "a" and a situation referring expression, an idiomatic schema is retrieved.

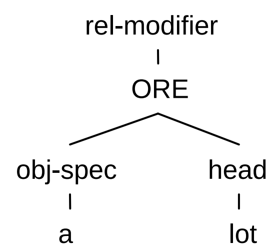

Figure 14: an idiomatic schema

This schema is idiomatic in that the object referring expression "a lot" is recognized as an expression which functions as a relation modifier despite its nominal form. (In my youth, I assumed that "a lot" was the single word "alot" which I used like a normal adverb until my grammar teacher corrected my error.) This schema is integrated with the preceding context giving:

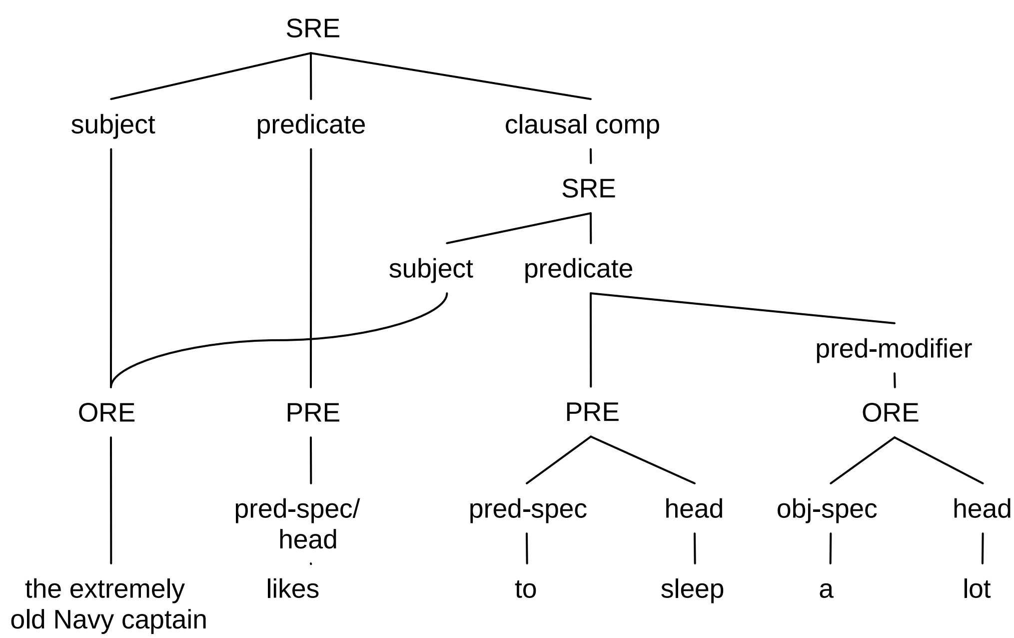

Figure 15: a fully elaborated situation referring expression

Note the decision to integrate the "a lot" schema with "to sleep" rather than "likes" in this example.

The processing of the input text is now complete and a representation of the referential and relational meaning of the expression—a situation referring expression describing a situation of "liking to sleep a lot" involving an "extremely old Navy captain"—is available in short-term working memory for use in subsequent processing.

During the processing of this sentence, there were several decision points at which alternative processing decisions could have been made (delaying the integration of the subject with "likes" until after the processing of the object, delaying the integration of a head noun until the lexical item following the noun is identified and determined not to be another noun). There is likely to be large variation across individuals (and even within individuals in different contexts) in terms of the selection of schemas and the timing of their integration. It may also be true that in contexts where one is only skimming a text, the integration process may be largely circumvented, resulting in only partially integrated representations. In general, it is assumed that humans learn **effective processing strategies** that tend to work efficiently given their goals and objectives, but that are subject to modification in particular contexts or when those goals and objectives change. For example, for verbs that take either an object or a clausal complement ("I believe him" vs. "I believe he is a hero"), the frequency of occurrence of objects and clausal complements will contribute to determining which schema is initially preferred. However, frequency is not the only factor which influences schema selection. It may be more disruptive to the processing mechanism to have to recover from the inappropriate integration of an object referring expression as the object of a clausal complement verb, then to delay the elaboration of the object until it can be determined that the object referring expression is not the subject of a clausal complement. There is some evidence that experienced readers avoid such early commitments and are less likely to be "garden-pathed" when the subsequent context indicates the early commitment was incorrect. True garden-pathing, where readers are forced to backup and re-read a piece of text, is assumed to be uncommon and typically involves restarting from near the beginning of the text, carrying along contextual information, rather than any kind of formal backtracking. More usually, representations are modified to accommodate inputs which are inconsistent with the prior context rather than being deconstructed and rebuilt.

The preceding example considered the processing of a sample sentence into a representation of its referential and relational meaning. In conjunction with the creation of a representation of the text, a situation model [16] is also constructed. The creation of the situation model corresponding to a text is driven by the grounding of the referring expressions of the text being processed. That is, whenever a referring expression is recognized, the situation model is updated to reflect the occurrence of that referring expression. Thus, referring expressions are crucial to the processing mechanism as well as the representational system. The construction of a situation model to ground the referring expressions in a text is an active area of research [1]. Construction-Integration Theory [16] and the Immersed Experiencer Framework [20] may provide much of the theoretical framework for this important component of Double R Process.

4. ACT-R

Double R Model is implemented in ACT-R 5.0 [22,23]. ACT-R is a cognitive architecture and modeling environment for the development of computational cognitive models. ACT-R has been used extensively in the modeling of higher-level cognitive processes (see the ACT-R web site at http://act-r.psy.cmu.edu/ for an extensive list of models and publications). ACT-R includes symbolic **production** and **declarative memory** systems integrated with subsymbolic **production selection** and **spreading activation** and **decay** mechanisms. Production selection involves the parallel matching of the left-hand side of all productions against a collection of **buffers** (goal buffer, retrieval buffer, visual buffer, auditory buffer) which contain the active contents of memory and perception. Production execution is a serial process—only one production is executed at a time. The parallel spreading activation and decay mechanism determines which declarative memory chunk is put into the retrieval buffer for comparison against productions. With its symbolic and subsymbolic processing mechanisms, ACT-R is a hybrid system of cognition. The **noise** parameter used by these computational mechanisms adds stochasticity to the system. ACT-R supports **single inheritance** of declarative memory chunks, limited, variable-based **pattern matching** (including a **partial-matching** capability), and **forward chaining**. ACT-R incorporates **learning** mechanisms for learning both declarative and procedural knowledge. ACT-R includes a **perceptual-motor component** supporting the development of embodied cognitive models. With the addition of the perceptual-motor component, and the use of buffers as the interface between various cognitive modules (e.g. vision module, auditory module, production system, declarative memory), ACT-R is referred to as an "integrated theory of the mind" [23].

5. Double R Model

Double R Model is currently capable of processing an interesting range of grammatical constructions including: intransitive, transitive and ditransitive verbs; verbs taking clausal complements; predicate nominals, predicate adjectives and predicate prepositions; conjunctions of numerous grammatical types; modification by attributive adjectives, prepositional phrases and adverbs, etc. Double R Model accepts as input as little as a single word or as much as an entire chunk of discourse—using the perceptual component of ACT-R to read words from a text window. Unrecognized words are simply ignored. Unrecognized grammatical forms result in partially analyzed text, not failure. The output of the model is a collection of declarative memory chunks that represent the referential and relational meaning of the input text. The code for version 1 of the model is available on the Double R Theory web site at www.DoubleRTheory.com.

While Double R Model can handle the basic sentence types described in [24], it does not yet handle such things as relative clauses, auxiliary inversion in question formation, and other non-canonical sentence forms. The initial application of Double R Model is for the modeling of pilot comm and many of these non-canonical sentence types are uncommon in that domain. However, question forms are important and Double R Model needs to be extended to handle them. Although Double R Model does not handle non-canonical sentence forms, it is robust in the sense of not crashing on ungrammatical input or on texts for which it lacks the appropriate schemas—an important requirement for handling pilot comm.

6. A Brief Comparison with Anderson, Budiu and Reder

Anderson, Budiu and Reder [25] (henceforth AB&R) present a theory of sentence processing within the ACT-R architecture in the context of memory retention. The focus of AB&R is on the real-time modeling of the empirical results of six different memory retention experiments. AB&R adopt a minimalist approach, severely limiting the amount of processing and inferencing that occurs during language processing. They claim that “constraints on processing time force the models in the direction of minimalist encoding.” That is, given the rapidity with which humans process language, there just isn’t time for a lot of extraneous processing and inferencing to occur. In fact, the only inference they allow involves the identification of the referent of a sentence. Further, AB&R claim that “there is nothing special about sentence memory.” Memory for sentences is no different than memory for other kinds of input.

AB&R posit three different types of representations that get created during language processing: a representation of surface structure, a propositional representation, and a situation representation limited to a single referent. These representations are composed of two types of declarative memory chunks: nodes, and links between nodes. AB&R claim that surface structure representations are more complex and have more links than propositional representations. Representations with more links are more difficult to retrieve from, or retain in, memory than representations with fewer links. This explains the poorer memory for surface structure without positing different retention functions. Further, referents are deeply encoded chunks that are more accessible than surface or propositional representation chunks. The models AB&R develop demonstrate the differential retention of surface, propositional and situational information, matching human data, without positing different retention functions as in numerous other theories. AB&R’s success in modeling the results of these six experiments is impressive and provides support for the cognitive validity of the underlying ACT-R architecture.

In Double R Theory there are only two levels of representation: linguistic (semantic) and situation model. There are no purely abstract concepts which could form a semantic level of representation distinct from linguistic representations. Although AB&R provide empirical support for the existence of structural representations, on their own minimalist assumption, it is difficult to see why a separate structural representation that does not aid comprehension would be constructed.

A key element of the AB&R models is the use of chunks corresponding to the links in their representations. Double R Model does not currently use chunks corresponding to links (although this approach is under serious consideration), instead using slots within chunks to encode associations. The use of link chunks has the advantage of externalizing associations, allowing them to be accessed directly and overcoming a problem in frame based systems of having to search into frames to find associations. That is, before you can find the association, you must first find the chunk for an associate and look for the association within that chunk. The use of externalized links is particularly important in ACT-R because of the single level spread of activation from the slots in the goal chunk to declarative memory. There is no mechanism for indirectly activating chunks via multi-level activation spread as in C-I Theory [16], necessitating very bushy representations in ACT-R. For activation to spread to all the links in a representation, there must be an index that ties the links together. AB&R use a context slot with a shared value for this purpose. In their example "Bob paid the waiter", where the sentence corresponds to a single proposition, tying the links together via the context slot works well, but it is unclear how this will work in more complex examples with multiple propositions in a single sentence. Will the context slot have different values for each proposition, similar to the way C-I Theory [16] represents propositions (relying on argument overlap to tie propositions together), or will the propositions be tied together more globally? AB&R's discussion of the restaurant script suggests that multiple propositions will be tied together via the context chunk, even across multiple sentences. However, the empirical consequences of this are not explored since the models of the six experiments all use very simple sentences.

AB&R posit the determination of a single referent for the propositional representation of a sentence. This referent is determined at the end of processing of the sentence, as an initial simplification. It corresponds to some deeply entrenched proposition in memory. In their example "Bob paid the waiter" the referent that is identified corresponds to an abstract proposition of "some person paying some waiter" embedded within a restaurant script with other associated propositions. In Double R Process, four distinct referring expressions would be identified "Bob" (ORE), "the waiter" (ORE), "paid" (PRE) and "Bob paid the waiter" (SRE), each of which would drive identification or creation of a referent corresponding to the specific object and action being referred to and not to the identification of some abstract proposition for the entire sentence.

In AB&R, the processing of the word "Bob" leads to the creation of a sentence and proposition node. In Double R Process, only an object referring expression would be created (and its referent identified). No expectation for the occurrence of a sentence or proposition would be established, since object referring expressions are non-relational and don't typically establish such expectations.

In AB&R, the basic processing mechanism is left-to-right, incremental, interactive (although there are distinct syntactic and propositional representations, they are created in parallel) and lexically driven (with some top-down influences). AB&R make use of the basic pattern matching and forward chaining capabilities of ACT-R. During processing, structures may be modified to accommodate the evolving context without backtracking. Double R Process makes similar processing commitments and Double R Model relies on similar processing features of ACT-R. Despite the very different focus of AB&R and Double R Model, i.e. real-time modeling of experimental results vs. creation of functional language comprehension systems, AB&R's theory of sentence processing and Double R Process are quite compatible—with the existence of structural representations and abstract concepts and propositions in AB&R representing the biggest differences.

7. Summary and Future Research

Double R Model may be the first attempt at the development of a functional language comprehension system founded on the principles of Cognitive Linguistics and implemented in the ACT-R cognitive modeling environment. Much work remains to be done. Double R Model has not yet reached a scale at which it can handle more than a subset of English. To expand the symbolic capabilities of Double R Model we are evaluating the integration of the CYC knowledge base [26,27], MikroKosmos [28], WordNet [29], and FrameNet [30]. CYC, annotated with gifs to represent the perceptual experience of concepts, could provide the basis for creation of a situation model to ground the referring expressions in a text, thereby, supporting a fuller representation of referential meaning. WordNet will support the expansion of the lexicon to a full complement of lexical items. MikroKosmos could provide a lexical semantics to go with Double R Grammar's grammatical semantics. FrameNet, with some mapping to Double R Grammar, could provide constructional schemas for relational and referential lexical items. To expand the subsymbolic capabilities of Double R Model (e.g. in support of lexical disambiguation), we are evaluating the use of Latent Semantic Analysis (LSA) [31], and considering modifications to ACT-R's single-level spreading activation mechanism, or incorporating link chunks into Double R Model. In this regard, LSA might provide an empirical basis for determining the strength of association of declarative memory chunks, and multiple-level spreading activation (like that proposed in ACT* [17]) would reduce the need for direct association of all related declarative memory chunks.

7. Acknowledgments

I would like to thank Kevin Gluck and Wink Bennett of the Air Force Research Laboratory for their support of this research and Col (s) Stuart Rodgers for collaborating on the integration of the language model with CYC.

8. References

[1] Ball, J. (2003). Double R Grammar. http://www.DoubleRTheory.com/DoubleRGrammar.pdf

[2] Langacker, R. (1987). *Foundations of Cognitive Grammar, Volume 1, Theoretical Prerequisites*. Stanford, CA: Stanford University Press

[3] Langacker, R. (1991). *Foundations of Cognitive Grammar, Volume 2, Descriptive Applications*. Stanford, CA: Stanford University Press.

[4] Talmy, L. (2003). *Toward a Cognitive Semantics, Vols I and II.* Cambridge, MA: The MIT Press.

[5] Lakoff, G., & M. Johnson (1980). *Metaphors We Live By*. Chicago: The University of Chicago Press.

[6] Rosch, E. (1978). Principles of Categorization. In *Cognition and Categorization*. Edited by E. Rosch & B. Lloyd. Hillsdale, NJ: LEA

[7] Wittgenstein, L. (1953). *Philosophical Investigations*. New York: MacMillan.

[8] Lakoff, G. (1987). *Women, Fire and Dangerous Things*. Chicago: The University of Chicago Press

[9] Chomsky, N. (1970). Remarks on Nominalization. In Jacobs & Rosembaum, eds., *Readings in English Transformational Grammar.* Ginn, Waltham, MA.

[10] Ball, J (2003). Towards a Semantics of X-Bar Theory. http://www.DoubleRTheory.com/SemanticsOfXBarTheory.pdf

[11] Cheng, L, and Sybesma, R. (1998). Interview with James McCawley, University of Chicago. *Glot International* 3:5, May 1998.

[12] Ball, J. (2003). "Is the Head of a Noun Phrase necessarily a Noun." Presentation at ICLC 2003. http://www.DoubleRTheory.com/ICLCPresentation.pps

[13] MacWhinney, B (1987). Toward a psycholinguistically plausible parser. In *ESCOL 1986*. Edited by S. Thomason. Ohio State University: Columbus, OH.

[14] Townsend, D. and T. Bever (2001). *Sentence Comprehension*. Cambridge, MA: The MIT Press.

[15] Ericsson, K & Kintsch, W. (1995). Long-Term Working Memory. *Psychological Review*, 102:2 pp. 211-245.

[16] Kintsch, W. (1998). *Comprehension, a Paradigm for Cognition*. New York, NY: Cambridge Univ Press.

[17] Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

[18] Collins, A. & E. Loftus (1975). A Spreading-Activation Theory of Semantic Processing. *Psychological Review*, 82, pp. 240-248.

[19] Wilks, Y. (1975). Preference Semantics. In *Formal Semantics*. Edited by E. Keenan. NY: Cambridge University Press.

[20] Zwann, R. (to appear). *The Psychology of Learning and Motivation*, Vol 44. Edited by B. Ross. New York: Academic Press

[21] Gernsbacher, M. A. (1990). *Language comprehension as structrure building*. Hillsdale, NJ: LEA.

[22] Anderson, J. & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahway, MJ: LEA

[23] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S, Lebiere, C, and Qin, Y.. (in press). An Integrated Theory of the Mind. *Psychological Review*.

[24] Quirk, R., S. Greenbaum, G. Leech, & J. Svartvik (1985). *A Comprehensive Grammar of the English Language*. London: Longman.

[25] Anderson, J. R., Budiu, R., and Reder, L. (2001). A Theory of Sentence Memory as Part of a General Theory of Memory. *Journal of Memory and Language*, 45, 337-367.

[26] Lenat et al. (2003). CYC. http://www.cyc.com

[27] Ball, J., Rodgers, S., & Gluck, K. (submitted). Integrating ACT-R and Cyc in a large-scale model of language comprehension for use in intelligent agents.

[28] Nirenburg, S. & Raskin, V. (to appear). *Ontological Semantics*. Cambridge: The MIT Press.

[29] Miller et al. (2003). WordNet. http://www.cogsci.princeton.edu/~wn/

[30] Fillmore et al. (2003). FrameNet http://www.icsi.berkeley.edu/~framenet/

[31] Landauer et al. (2003). Latent Semantic Analysis (LSA). http://lsa.colorado.edu/

Author Bibliography

JERRY T. BALL is a Senior Research Psychologist working for the Air Force Research Laboratory, Human Effectiveness Directorate, Warfighter Training Research Division in Mesa, AZ. Dr. Ball earned a PhD in Cognitive Psychology from New Mexico State University and a Masters Degree in Computer and Information Science from the University of Florida. His research interests include the development of cognitively plausible language comprehension systems and the modeling of human behavior in complex environments.